Phys 396 2018

Lecture 20

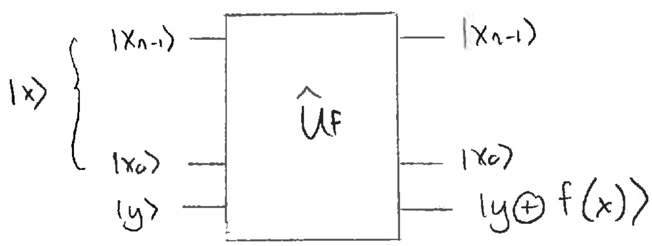
Tues: HW due

Oracle queries

Consider a classical function that maps n bits onto a single bit

$$f : \underbrace{(x_{n-1}, \ldots, x_0)}_{x} \rightarrow f(x)$$

This can be implemented via a unitary oracle

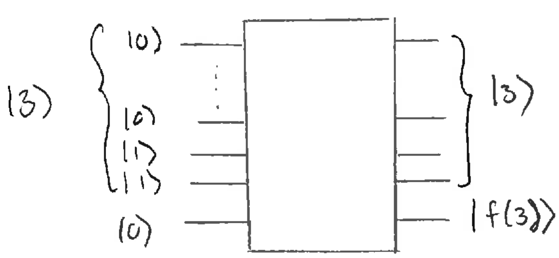


or $$\hat{U}_f |x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$$

n qubits 1 qubit

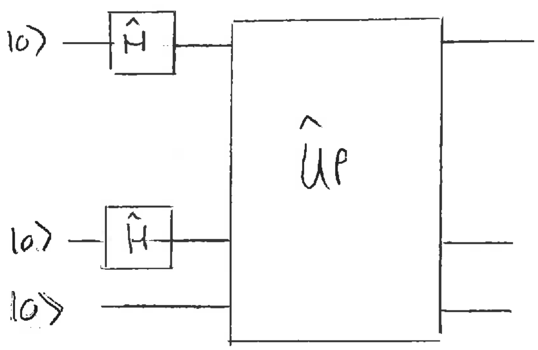
computational basis

Then classical function evaluation is done by supplying a single computational basis state to the oracle. For example



$\rightsquigarrow$ measure in comp basis $\rightsquigarrow f(3)$

We saw that quantum physics allows us to use states that are superpositions of computational basis states. Specifically we saw that



This produces the state

$$\frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1} |x\rangle|0\rangle$$

immediately after the Hadamards and then the state

$$\frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1} |x\rangle|f(x)\rangle$$

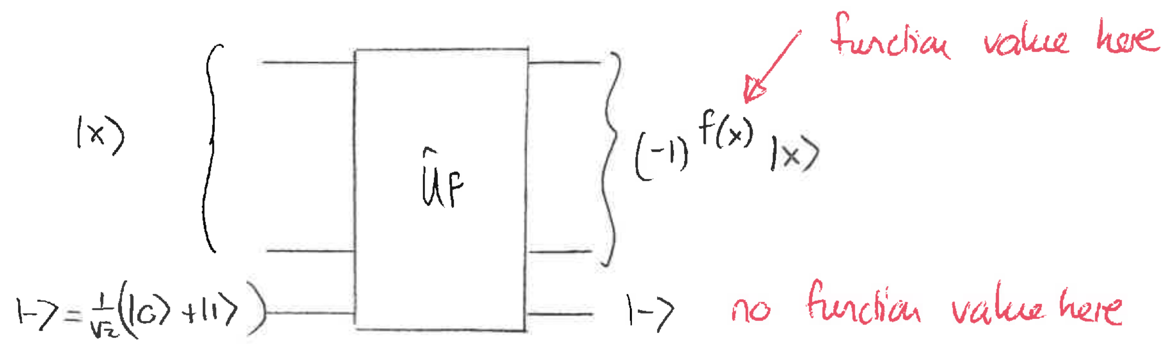
immediately after the oracle. So this has somehow accessed f evaluated at all possible inputs with just one oracle invocation.

So

> The existence of superpositions in quantum physics allows for a type of simultaneous function evaluation across all possible function arguments. This is some form of parallelism

Can we somehow harness this to learn global properties of f. This is not immediately obvious. Clearly computational basis measurements will not help

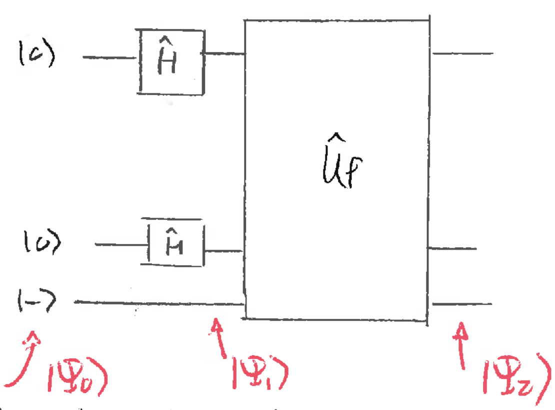
Separately we saw that if we use a superposition in the function register, then this pushes function evaluation into a phase over the argument register.



So

$$\boxed{\hat{U}_f |x\rangle|-\rangle = (-1)^{f(x)} |x\rangle|-\rangle.}$$

We can now consider the combination of these two superposition strategies



If the initial state is

$$|\Psi_0\rangle = |0\ldots0\rangle|-\rangle$$

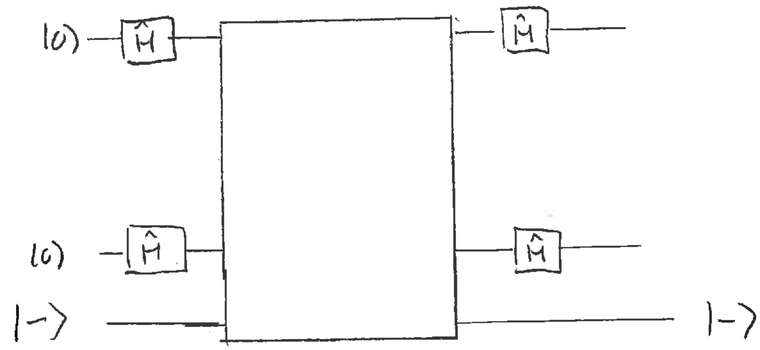
then

$$|\Psi_0\rangle \xrightarrow{\hat{H}\otimes\ldots\otimes\hat{H}\otimes\hat{I}} \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle|-\rangle \equiv |\Psi_1\rangle$$

Then

$$|\Psi_2\rangle = \hat{U}_f|\Psi_1\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle|-\rangle$$

Perhaps at this point a measurement on the argument register yields some information about f. But a computational basis measurement will clearly just give one of the argument values and this again is no help. We might try measurements in other bases. Specifically we will consider:



In order to assess this we need an algebraic description of the Hadamard.

1 Hadamard transformation

a) Consider a Hadamard transformation on a single qubit. Show that for $x = 0, 1$

$$\hat{H}\left|x\right\rangle = \frac{1}{\sqrt{2}}\sum_{y=0}^{1}(-1)^{xy}\left|y\right\rangle.$$

b) Consider Hadamard transformations on n qubits. Show that

$$\hat{H}\otimes\cdots\otimes\hat{H}\left|x_{n-1}\dots x_0\right\rangle = \frac{1}{2^{n/2}}\sum_{y_{n-1},\dots,y_0=0}^{1}(-1)^{x_{n-1}y_{n-1}+\cdots+x_0y_0}\left|y_{n-1}\dots y_0\right\rangle.$$

Answer

a) Know $\hat{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

$\hat{H}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

The formula says

$$\hat{H}|0\rangle = \frac{1}{\sqrt{2}}\sum_{y=0}^{1}(-1)^{0y}|y\rangle = \frac{1}{\sqrt{2}}\sum_{y=0}^{1}|y\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$\hat{H}|1\rangle = \frac{1}{\sqrt{2}}\sum_{y=0}^{1}(-1)^{1y}|y\rangle = \frac{1}{\sqrt{2}}\sum_{y=0}^{1}(-1)^{y}|y\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

It's correct.

b) $\hat{H}\otimes\dots\otimes\hat{H}|x_{n-1}\dots x_0\rangle = H|x_{n-1}\rangle \dots H|x_0\rangle$

$$= \frac{1}{\sqrt{2}}\sum_{y_{n-1}}(-1)^{x_{n-1}y_{n-1}}|y_{n-1}\rangle \dots \frac{1}{\sqrt{2}}\sum_{y_0}(-1)^{x_0y_0}|y_0\rangle$$

$$= \frac{1}{2^{n/2}}\sum_{y_{n-1}\dots y_0}(-1)^{x_{n-1}y_{n-1}}\dots(-1)^{x_0y_0}|y_{n-1}\dots y_0\rangle$$

$$= \frac{1}{2^{n/2}}\sum_{y_{n-1}\dots y_0}(-1)^{x_{n-1}y_{n-1}+\dots+x_0y_0}|y_{n-1}\dots y_0\rangle$$

This gives a useful general rule

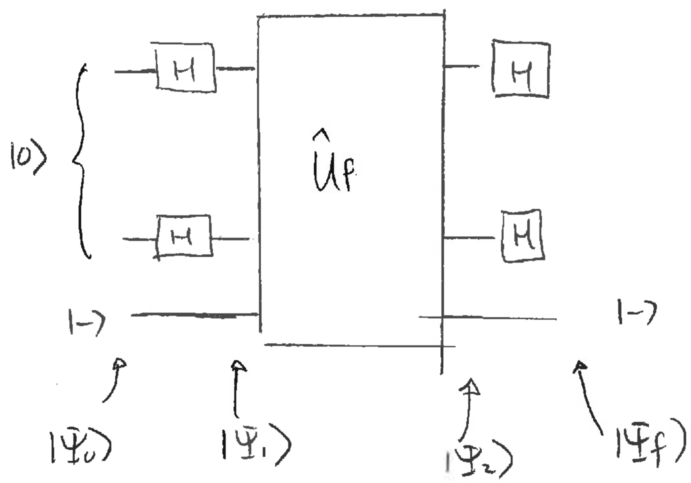
$$\boxed{\hat{H} \otimes \cdots \otimes \hat{H} \, |x\rangle = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} (-1)^{x\cdot y} \, |y\rangle}$$

n qubits n qubits

where $x \cdot y$ is a discrete inner product of two n bit numbers:

$$x \cdot y = x_{n-1}y_{n-1} \cdots x_0 y_0$$

Thus our circuit



produces

$$|\Psi_f\rangle = \frac{1}{2^{n/2}} \frac{1}{2^{n/2}} \sum_x \sum_y (-1)^{f(x)} (-1)^{x\cdot y} \, |y\rangle$$

Now suppose that we perform a computational basis measurement

2 Global function evaluation

After the circuit described in class, the state of the system is

$$|\Psi_3\rangle = \frac{1}{2^n}\sum_x\sum_y (-1)^{f(x)}(-1)^{x\cdot y}|y\rangle .$$

a) Determine an expression for the probability with which a computational basis measurement yields the outcome $00\dots0 \equiv 0$.

b) Suppose that the function is constant. Determine the probability with which a computational basis measurement yields the outcome $00\dots0 \equiv 0$.

Answer: a) $\text{Prob}(0) = |\langle 0\dots0|\Psi_F\rangle|^2$

So $\langle 0\dots 0|\Psi_F\rangle = \frac{1}{2^n}\sum_x\sum_y(-1)^{f(x)}(-1)^{x\cdot y}\langle 0|y\rangle$ ← only non-zero when $y \neq 0$

$= \frac{1}{2^n}\sum_x(-1)^{f(x)}$

$\text{Prob}(0) = \left(\frac{1}{2^n}\right)^2\left|\sum_x(-1)^{f(x)}\right|^2$

b) There are two possibilities First for $f=0$

$\text{Prob}(0) = \left(\frac{1}{2^n}\right)^2\left|\underbrace{\sum_x 1}_{2^n}\right|^2 = 1$

Then for $f=1$

$\text{Prob}(0) = \left(\frac{1}{2^n}\right)^2\left|\underbrace{\sum_x(-1)}_{-2^n}\right|^2 = 1$

This looks like it can determine whether the function that determines the oracle is constant. We have:

supply f $\rightarrow$ construct $\hat{U}_F$ $\rightarrow$ Use oracle once in circuit

$\hookrightarrow$ if f constant get 0 with certainty.

So we know that if the computational basis measurement gives an outcome $\neq 0$, then the function is not constant.

Is there some category of function for which one gets a non-zero outcome?

Deutsch-Josza algorithm

Consider functions taken from one of two categories:

category 1	category 2
f is constant $\Rightarrow$ gives same outcome for all inputs	f is balanced $\Rightarrow$ gives 0 for exactly half inputs and 1 for exactly half inputs.

3 Deutsch-Jozsa algorithm

a) Consider the following functions of two bits

$$f_1(x_2,x_1)\ f_1 = x_1$$
$$f_2 = x_2$$
$$f_3 = x_2 \oplus x_1$$
$$f_4 = x_2 x_1$$

Determine which of these are balanced.

b) Suppose that you are given an oracle corresponding to either an n bit constant or a balanced function. How many classical oracle queries does it require to determine which type of function you are given?

c) Consider a general n bit balanced function. Determine the probability with which the algorithm in class yields a computational basis measurement outcome of 0.

Answer:

a)

x_2	x_1	f_1	f_2	f_3	f_4
0	0	0	0	0	0
0	1	1	0	1	0
1	0	0	1	1	0
1	1	1	1	0	1
		Yes	Yes	Yes	No

b) If it is balanced then enquiring on $2^n/2 + 1$ function arguments will certainly reveal this. Need $2^{n-1}+1$ oracle queries

c) $\text{Prob}(0) = \left(\frac{1}{2^n}\right)^2 \left| \sum_x (-1)^{f(x)} \right|^2 = 0$

get 1 half time, -1 half time

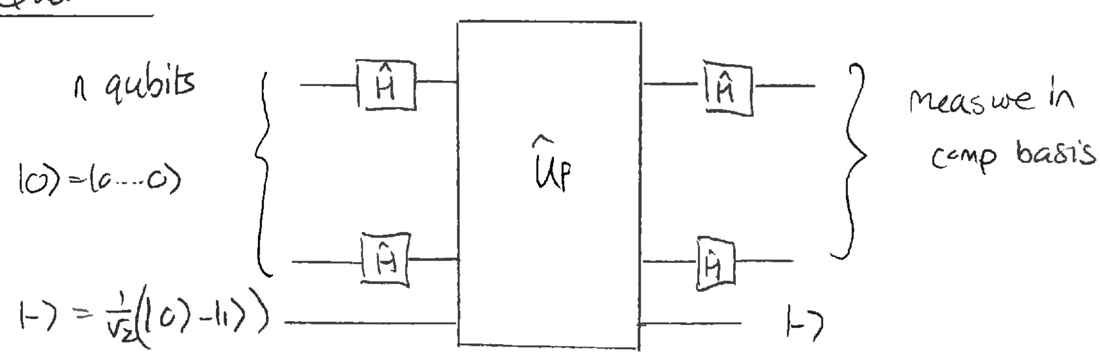
$\text{Prob}(0) = 0$

This gives a quantum algorithm for solving the Deutsch-Jozsa problem:

Problem: f is a function of n bits that is either constant or else balanced. Determine which it is with minimal oracle evaluations

Classical: Using $2^{n-1}+1$ oracle queries reveals whether f is balanced or constant with certainty.

Quantum:



If f is constant comp basis measurement $\rightarrow$ 0 with certainty

If f is balanced " " " $\rightarrow$ never gives 0.

Thus

With a single oracle invocation one can solve the DJ problem with certainty. Exponential speed up in oracle queries